What I claim as my invention is:

1.   A method of translating addresses of a predetermined length, comprising steps of:

5   (1)   resolving the address to a prefix, if the address is found in a primary translation table, the primary translation table containing prefixes whose lengths are less than a predetermined value less than the length of the address and locations of branch data structures a plurality of secondary search units;

      (2)   performing a secondary search in the secondary search units in parallel, if

10   the primary translation table indicates the locations of branch data structures to begin each secondary search for prefixes whose lengths are larger than the predetermined value; and

      (3)   translating the addresses to prefixes, if the prefixes are found in the secondary search.

15

2.   The address translation method according to claim 1, comprising a further step of:

      selecting one of a plurality of secondary search units in response to the location of a branch data structure.

20

3.   The address translation method according to claim 2, comprising a further step of:

      buffering addresses to be translated at the secondary search units for secondary searches so that a plurality of addresses are translated in parallel in an

25   orderly fashion.   .

4.   The address translation method according to claim 3, comprising a further step of:

      scrambling the address according to a predetermined reproducible formula.

30

5.   The method as claimed in claim 2 wherein the selection of the secondary search units is performed according a round-robin discipline.

6.   A method of encoding a number H>0 of independent trees of known

35   prefixes and respective prefix translations, the encoded trees being stored in a

single memory, the method constructing a data structure having a table V storing pointers and a table T storing translations wherein each of the H encoded trees is identified by a root with corresponding entries in tables V and T.

5   7.   The method as claimed in claim 6 wherein the tables V and T are stored in two separate memories the entries of which bearing one-to-one correspondence to each other.

8.   The method as claimed in claim 6 wherein the encoded independent trees
10   are interleaved so that successive entries in the tables V and T may belong to different trees.

9.   The method as claimed in claim 6 wherein the search algorithm of any tree h, $0<h\leq H$, having a root identified by index h in tables V and T, includes the steps:
15   setting $K=H+F$ and $y=h+F$, F being an arbitrary integer offset representing a reference memory address,
        initializing tables V and T by zero entries, and:

for $1\leq j\leq M$
    {
20   *for* $1\leq d<m_j$
        { $x=D_{j,d}$
            if $(V(x, y)>0)$, $y=V(x, y)$;
            else $\{K\rightarrow K+1,\ V(x, y)=K,\ y=K\}$
        }
25   *for* $d=m_j, x=D_{j,d}, T(x, y)=G$
    }

m_j being the number of bits in jth prefix in a prefix directory, and G being the sought translation.

30

10. The method as claimed in claim 6 wherein the decoding of an arbitrary address to retrieve a respective translation follows the algorithm:

u_j is the jth bit of a B-bit received address, h is the respective tree number, y=h, t=0,

for $1 \leq d \leq B$

{

         $x=u_d$; if (y=0) exit;

         if (T(x, y)>0) t=T(x, y));

         y=V(x, y);

         if (V(x, y)>0) y=V(x, y)

}

wherein if t greater than zero, it is a valid translation, and if t is zero the address is declared unknown.

11. The method as claimed in claim 6 further comprising step of complementing the known prefixes before encoding.

12. The method as claimed in claim 6 wherein the number of trees per secondary memory is less than an upper bound

13. A method of resolving B bit long addresses of packets into prefixes of any length up to B by the use of a data structure which comprises a length sorted table Q and a plurality of secondary search units, table Q containing data related to prefixes of length less than A, A≤B and each secondary search units including tables V and T which are in one-to-one correspondence to one another and each consists of a 2xM memory, M being a positive integer, comprising steps of:

    (1) indexing table Q by using the first A bits of an address to generate a corresponding prefix of length equal to or less than A, or a pointer to a secondary search unit;

    (2) accessing table V of the secondary search unit indicated by the pointer using each successive remaining bit of the address in order;

    (3) accessing table T of the secondary search unit at each successive location corresponding to the location of table V accessed in step (2); and

    (4) reading a valid data contained at the location in table T, the valid data

being a prefix of length more than A.

14.   The method according to claim 13 wherein steps (2)-(4) are performed in one or more of the secondary search units in parallel.

5

15.   The method according to claim 14, wherein each secondary unit comprises one or more search branches and a pointer to a secondary search unit indicates the identity of the secondary unit and a branch number h so that data relating to two or more search branches are interleaved in tables V and T of a secondary unit, the

10   method further comprises a step of accessing table V at a location indicated by the branch number h of the secondary unit identified by the pointer.

16.   The method according to claim 15, wherein the step of accessing table V in each identified secondary search unit comprises further steps of:

15   (5) accessing table V by using the first bit of the remaining bits of the address at a location identified by branch number h;

(6) reading table V at the location which indicates a next location of table V;

(7) continue accessing and reading each successive location using each

20   successive bit of the remaining bits until the Bth bit; and

(8) continue reading the valid data contained at the location in table T corresponding to the successive locations read in the step (7).

17.   The method according to claim 16, comprising further steps of:

25   accessing table T at a location corresponding to each successive location accessed in steps (5)-(7);

recording the data contained at each accessed location of table T; and

reading the valid data contained at the latest previously accessed location if the last accessed locations contained no valid data.

30

18.   The method according to claim 17, wherein u_j is the jth bit of the B-bit received address, and the address is known to belong to branch h, the steps of accessing and reading table V and T in each secondary search unit are performed as follows to generate a prefix t:

35

```
y=h, t=0,
for 1≤d≤B
{
        x=u_d ;
        if (T(x, y)>0)  t=T(x, y) );
        y=V(x, y)
        if V(x, y)>0;   y=V(x, y)
}
```

19.   The method according to claim 18, wherein the step of reading the valid data contained at the location in table T is replaced by steps of:

reading an indicator contained at the location in table T;

accessing a storage device at a location indicated by the indicator; and

reading the valid data contained at the location in the storage device.

20.   The method according to claim 19, further comprising a step of:

scrambling all the bits of the addresses to be translated according to a reproducible formula.

21.   The method according to claim 20, wherein the reproducible formula is a bit reversal of all the bits in the addresses.

22.   The method according to claim 13, wherein length sorted table Q provides an indication of '00' if the address cannot be resolved, "01" if the prefix is found, and "10", if a further search is required in a respective secondary search unit in that a "10" outcome is associated with a branch number.

23.   An apparatus for address translation of a packet, comprising:

a parsing block for receiving the packet and parsing address, each address having length B, B being a positive integer;

an indexing block for selecting the first A binary bits of each received address, A being a predetermined positive integer and A≤B and for directly accessing a sorted prefix directory by the first A binary bits, the sorted prefix directory containing translated prefixes of length N equal to or shorter than A and data specifying one of a plurality of secondary search units;

the plurality of secondary search units having the plurality of secondary memories for searching in parallel through the secondary memories specified by the indexing block for prefixes of length N longer than A, each secondary memory comprising tables V and T in that tables V and T are in a one-to-one

5 correspondence to one another and each consists of a 2xM memory, M being a positive integer, table V for accessing successive location for each successive bits above A of the addresses and table T for translated prefixes at a location corresponding to the location accessed in table V.

10 24. The apparatus for address translation of a packet according to claim 23, wherein each prefix of length longer than A belong to any of prefix trees and each secondary memory containing prefixes of one or more prefix tree.

25. The apparatus according to claim 24 wherein tables V and T of each
15 secondary memory have data structures containing translated addresses J for addresses belonging to address tree branch h and $1 \leq h \leq H$, $K=H+F$, $y=h+F$, F being an arbitrary integer offset representing a reference memory address in that J is generated as follows:

20 *for* $1 \leq j \leq M$
    {
        for $1 \leq d < m_j$
        { $x=D_{j,d}$
        if $V(x, y) > 0$, $y=V(x, y)$
25   *else* $\{K \rightarrow K+1,$ $V(x, y)=K,$ $y=K\}$ }
        for $d=m_j,$ $x=D_{j,d},$ $T(x, y)=J$
    }

wherein a plurality of addresses are sorted in a sorted address directory R, the jth address in position j in directory R has m_j bits; the bit in position d, $0 \leq d < m_j$, in the
30 j^{th} index, is denoted $D_{j,d}$ and the value of $D_{j,d}$ is either "0" or "1".

26. The apparatus according to claim 25 further comprising a scrambling unit for scrambling the first A bits of the addresses to be translated according to a reproducible formula.

35

27. The apparatus according to claim 26 wherein the reproducible formula is a bit reversal of the first A bits in the addresses.

28. An address translation apparatus for telecommunications networks in
5 which packets are transported to addresses contained therein, comprising:

an address separation unit for separating from a packet an address to be translated;

primary translation unit having a primary translation table for translating the address to a prefix, the primary translation table containing prefixes whose
10 widths are less than a predetermined value and locations of branch search data structures in a secondary search units; and

a plurality of secondary search units for performing secondary searches in parallel, each secondary unit having the branch search data structure for performing each secondary search and translating the address to a prefix, if the
15 primary translation table indicates the location of a branch search data structure to begin the secondary search.

29. The address translation apparatus according to claim 28, further comprising:
20 address scrambling unit for scrambling the addresses according to a predetermined reproducible formula.

30. The address translation apparatus for telecommunications networks according to claim 29 further comprising:
25 a selector for selecting a secondary search unit for performing the secondary search.